

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for providing continuous data protection, the method comprising the steps of:

writing data to a primary volume;

duplicating the writes made to the primary volume to a secondary volume in a sequential fashion, wherein the secondary volume contains a chronological ordering of all writes made to the primary volume; and

identifying an any point in time (APIT) window wherein all writes to the secondary volume are maintained so that within the identified APIT window, the primary volume may be restored to any ~~previous~~ point within the APIT ~~window~~ window;

creating a mapping data structure to track changes in the data written to the secondary volume between two points in time; and

merging mapping data structures that protect contiguous points in time, whereby fewer mapping data structures are needed to protect the data between the same two points in time spanning the merged mapping data structures.

2. (Canceled)

3. (Previously presented) The method of claim 1 further including the step of retaining particular points in time beyond the APIT window.

4. (Previously presented) The method of claim 3 wherein a point-in-time map is created by creating a full mapping between the primary volume and the secondary volume for a point in time that is retained beyond the APIT window, the full mapping using a data structure.

5. (Previously presented) The method of claim 4 wherein the full mapping is created by merging data structures ranging in time from an initial time to a time when a snapshot was taken.

6. (Previously presented) The method of claim 4 wherein the full mapping is created by merging data structures ranging in time from a time a point-in-time map created prior to a snapshot was taken to a time when the snapshot was taken.

7. (Previously presented) The method of claim 1 wherein data on the secondary volume that is outside of the identified APIT window is discarded.

8. (Previously presented) The method of claim 1 wherein data on the secondary volume that is outside of the identified APIT window is phased out according to a retention policy.

9. (Previously presented) The method of claim 4 further comprising the step of periodically creating point-in-time maps to reduce a number of data structures that are needed when performing a restore.

10. (Currently amended) A method for operating a data protection system for a protected computer system, the method comprising the steps of:

tracking writes made to a primary volume;

duplicating the writes to the primary volume in a sequential fashion on a secondary volume, wherein the secondary volume contains a chronological ordering of all writes made to the primary volume;

organizing a mapping of the writes between the primary volume and the secondary volume into data structures, wherein the data structures enable the primary volume to be restored to any point in time; [[and]]

identifying an any point in time window wherein the data structures are maintained so that within the identified time window, the primary volume may be restored to any point within the time window; and

merging data structures that protect contiguous points in time, whereby fewer data structures are needed to protect the data between the same two points in time spanning the merged data structures.

11. (Previously presented) The method of claim 10 wherein a snapshot is taken at a particular point in time within the identified time window and a full mapping of the primary volume and the secondary volume for the particular point time is created.

12. (Previously presented) The method of claim 11 wherein the full mapping is created by merging data structures ranging in time from an initial time to a time when the snapshot was taken.

13. (Previously presented) The method of claim 11 wherein the full mapping is created by merging data structures ranging in time from a time when a point-in-time map created prior to the snapshot was taken to the time the snapshot was taken.

14. (Currently amended) A system for providing continuous data protection, the system comprising:

a host computer;

a primary volume for storing data written by the host computer;

a secondary volume wherein writes made to the primary volume are sequentially duplicated onto the secondary volume, the secondary volume containing a chronological ordering of all writes made to the primary volume; and

a data protection system configured ~~[[to]]~~ to:

manage the duplication of writes to the secondary ~~volume and to~~
volume;

map data between the primary volume and the secondary volume using data structures, wherein the data structures are maintained so that within an established time window, the primary volume may be restored to any point within the time window; and

merge data structures that protect contiguous points in time, whereby fewer data structures are needed to protect the data between the same two points in time spanning the merged data structures.

15. (Original) The system of claim 14 wherein the data protection system is configured to create a point-in-time map for a point-in-time at which the primary volume needs to be restored.

16. (Previously presented) The system of claim 15 wherein the point-in-time map is created by creating a full mapping between the primary volume and the secondary volume.

17. (Previously presented) The system of claim 16 wherein the full mapping is created by merging data structures ranging in time from an initial time to a time when a snapshot was taken.

18. (Previously presented) The system of claim 17 wherein the full mapping is created by merging data structures ranging in time from a time when a point-in-time map created prior to the snapshot was taken to the time the snapshot was taken.

19. (Canceled)

20. (Currently amended) A computer-readable storage medium containing a set of instructions for a general purpose computer, the set of instructions comprising:

a tracking code segment for tracking writes made to a primary volume;

a duplicating code segment for duplicating the writes to the primary volume in a sequential fashion on a secondary volume, wherein the secondary volume contains a chronological ordering of all writes made to the primary volume;

an organizing code segment for organizing a mapping of the writes between the primary volume and the secondary volume into data structures, wherein the data structures enable the primary volume to be restored to any point in time;
[[and]]

an identifying code segment for identifying an any point in time window wherein the data structures are maintained so that within the identified time window, the primary volume may be restored to any point within the time window;
and

a merging code segment for merging data structures that protect contiguous points in time, whereby fewer data structures are needed to protect the data between the same two points in time spanning the merged data structures.

21. (Currently amended) A method for enabling data recovery from a primary volume in a continuous data protection system, comprising the steps of:

creating an initial copy of the primary volume;

after the initial copy is made, duplicating each write to the primary volume to a write log to be stored on a secondary volume, the write log being written to the secondary volume when the write log is full, wherein the secondary volume contains a chronological ordering of all writes made to the primary volume; [[and]]

creating a snapshot of the primary volume, the snapshot being a point in time to which the primary volume can be ~~restored~~ restored;

creating a block-ordered data structure to track the writes made to the secondary volume, the data structure being derived from the write log and containing the writes made to the primary volume between two points in time; and

merging data structures that protect contiguous points in time, whereby fewer data structures are needed to protect the data between the same two points in time spanning the merged data structures.

22. (Canceled)

23. (Currently amended) The method according to claim [[22]] 21, wherein [[the]] creating [[step]] the snapshot includes inserting a marker into the write log, the marker indicating a time at which the snapshot is taken.

24. (Canceled)

25. (Currently amended) The method according to claim ~~24, further comprising the step of:~~ 21, wherein the merging step includes
~~optimizing the data structure by~~ merging data structures from a time corresponding to a snapshot to a current time.

26. (Currently amended) A method for providing continuous data protection, comprising the steps of:

writing data to a primary volume;

duplicating the writes made to the primary volume to a secondary volume in a sequential fashion, wherein the secondary volume contains a time-based ordering of all writes made to the primary volume; [[and]]

identifying an any point in time (APIT) window wherein all writes to the secondary volume are maintained so that within the identified APIT window, the

primary volume may be restored to any ~~previous~~ point within the APIT ~~window~~
window;

mapping the writes between the primary volume and secondary volume using
a data structure; and

merging data structures that protect contiguous points in time, whereby
fewer data structures are needed to protect the data between the same two points in
time spanning the merged data structures.

27. (Canceled)

28. (Previously presented) The method of claim 26, further comprising
the step of retaining particular points in time beyond the APIT window.

29. (Previously presented) The method of claim 28, wherein a point in
time map is created by creating a full mapping between the primary volume and the
secondary volume for a point in time that is retained beyond the APIT window, the
full mapping using a data structure.

30. (Previously presented) The method of claim 29, wherein the full mapping is created by merging data structures ranging in time from an initial time to a time when a snapshot was taken.

31. (Previously presented) The method of claim 29, wherein the full mapping is created by merging data structures ranging in time from a time a point in time map created prior to a snapshot was taken to a time when the snapshot was taken.

32. (Previously presented) The method of claim 29, further comprising the step of periodically creating point in time maps to reduce a number of data structures that are needed when performing a restore.

33. (Previously presented) The method of claim 26, wherein data on the secondary volume that is outside of the identified APIT window is discarded.

34. (Previously presented) The method of claim 26, wherein data on the secondary volume that is outside of the identified APIT window is phased out according to a retention policy.

35. (Currently amended) A computer implemented method for continuously protecting data stored on a volume of a storage system, comprising:

writing data to a primary volume in a block-based order;

duplicating the data to a write log in a sequential order;

creating a mapping data structure from the write log, the mapping data structure stored on a secondary volume and used to track changes in the data between two points in time, whereby the data is continuously protected; and

merging mapping data structures that protect contiguous points in time, whereby fewer mapping data structures are needed to protect the data between the same two points in time spanning the merged mapping data structures.

36. (Previously presented) The method according to claim 35, wherein the two points in time can be any points in time during which the data is protected.

37. (Previously presented) The method according to claim 35, further comprising:

creating an initial full copy snapshot of the primary volume such that changes to the data are relative to the data in the initial full copy snapshot.

38. (Previously presented) The method according to claim 35, further comprising:

creating a snapshot of the primary volume at a point in time by inserting a marker into the write log to identify the point in time.

39. (Canceled)